Cal.cpp

```
#include <iostream>
#include <stdio.h>
#include <stdlib.h>
#include <fstream>
#include <cstdlib>
#include <sys/time.h>
#include <stddef.h>

using namespace std;

int main()
{
	int datasamples = 50;
	int datasamples1 = 250;
	int x = 1;
	double data = 0;
	double gyroavearray [datasamples];
	double accelxarray [datasamples1];
	double accelyarray [datasamples1];
	double accelxavearray [datasamples];
	double accelyavearray [datasamples];
	char c[10];
	int counter = 1;
	FILE *file;
	FILE *file1;
	FILE *file2;
	FILE *file3;
	FILE *file4;
	FILE *file5;
	double gyrosum = 0;
	double gyroave = 0;
	double accelxmax = 0;
	double accelxmin = 0;
	double accelxave = 0;
	double accelxsum = 0;
	double accelymax = 0;
	double accelymin = 0;
	double accelyave = 0;
	double accelysum = 0;
	double convert = .004887585533;

	//Clear Files
	FILE *writefile1 = fopen("/mnt/cf/accelxcal1.txt", "w");
	fclose(writefile1);
	FILE *writefile2 = fopen("/mnt/cf/accelycal1.txt", "w");
	fclose(writefile2);
	FILE *writefile3 = fopen("/mnt/cf/gyrocal1.txt", "w");
```

```
fclose(writefile3);
FILE *writefile4 = fopen("/mnt/cf/accelxavecal.txt", "w");
fclose(writefile4);
FILE *writefile5 = fopen("/mnt/cf/accelyavecal.txt", "w");
fclose(writefile5);

//Get Data for Gyro
system("i2c-io 0x0b setdir b.4 out"); //Set Pin Direction
system("i2c-io 0x0b set b.4 0"); //Yellow LED on
cout << "Calibrating Gyro..." << endl;
while(x <= datasamples)
	{
		system("i2c-io 0x0b Get ADC.2 >> /mnt/cf/gyrocal1.txt");
		system("i2c-io 0x0b Get ADC.0 >> /mnt/cf/accelxavecal.txt");
		system("i2c-io 0x0b Get ADC.1 >> /mnt/cf/accelyavecal.txt");
		x++;
	}
system("i2c-io 0x0b set b.4 1"); //Yellow LED off


//Get Data for Accel X and Accel Y
x = 1;
cout << "Calibrating Accelerometers..." << endl;
while(x <= datasamples1)
	{
		system("i2c-io 0x0b set b.4 0"); //Yellow LED on
		system("i2c-io 0x0b Get ADC.0 >> /mnt/cf/accelxcal1.txt");
		system("i2c-io 0x0b Get ADC.1 >> /mnt/cf/accelycal1.txt");
		x++;
		system("i2c-io 0x0b set b.4 1"); //Yellow LED off
	}


//Averages Gyro Data
file = fopen("/mnt/cf/gyrocal1.txt", "r");
while(fgets(c, 10, file)!=NULL)
	{
		data = atof(c);
		gyroavearray[counter] = data;
		counter++;
	}
counter = 1;
while(counter <= datasamples)
	{
		gyrosum = gyrosum + gyroavearray[counter];
		counter++;
	}
gyroave = gyrosum / datasamples;
```

```
//Averages Accel X Data
file2 = fopen("/mnt/cf/accelxavecal.txt", "r");
counter = 1;
while(fgets(c, 10, file2)!=NULL)
	{
		data = atof(c);
		accelxavearray[counter] = data;
		counter++;
	}

counter = 1;
while(counter <= datasamples)
	{
		accelxsum = accelxsum + accelxavearray[counter];
		counter++;
	}

accelxave = accelxsum / datasamples;


//Averages Accel Y Data
file3 = fopen("/mnt/cf/accelyavecal.txt", "r");
counter = 1;
while(fgets(c, 10, file3)!=NULL)
	{
		data = atof(c);
		accelyavearray[counter] = data;
		counter++;
	}

counter = 1;
while(counter <= datasamples)
	{
		accelysum = accelysum + accelyavearray[counter];
		counter++;
	}

accelyave = accelysum / datasamples;




//Puts Accel X Data into accelxarray
file4 = fopen("/mnt/cf/accelxcal1.txt", "r");
counter = 1;
```

```
while(fgets(c, 10, file4)!=NULL)
	{
		data = atof(c);
		accelxarray[counter] = data;
		counter++;
	}

//Finding Max Values for Accel X
counter = 1;
accelxmax = accelxarray[1];
while(counter <= datasamples1)
	{
		if (accelxarray[counter] > accelxmax)
			{
				accelxmax = accelxarray[counter];
				counter++;
			}
		else
			counter++;
	}

//Finding Min Values for Accel X
counter = 1;
accelxmin = accelxarray[1];
while(counter <= datasamples1)
	{
		if (accelxmin > accelxarray[counter])
			{
				accelxmin = accelxarray[counter];
				counter++;
			}
		else
			counter++;
	}

//Puts Accel Y Data into accelyarray
file5 = fopen("/mnt/cf/accelycal1.txt", "r");
counter = 1;
while(fgets(c, 10, file5)!=NULL)
	{
		data = atof(c);
		accelyarray[counter] = data;
		counter++;
	}

//Finding Max Values for Accel Y
counter = 1;
accelymax = accelyarray[1];
while(counter <= datasamples1)
```

```
		{
			if (accelyarray[counter] > accelymax)
				{
					accelymax = accelyarray[counter];
					counter++;
				}
			else
				counter++;
		}

	//Finding Min Values for Accel Y
	counter = 1;
	accelymin = accelyarray[1];
	while(counter <= datasamples1)
		{
			if (accelyarray[counter] < accelymin)
				{
					accelymin = accelyarray[counter];
					counter++;
				}
			else
				counter++;
		}

	gyroave = gyroave*convert;
	accelxmax = accelxmax*convert;
	accelxmin = accelxmin*convert;
	accelxave = accelxave*convert;
	accelymax = accelymax*convert;
	accelymin = accelymin*convert;
	accelyave = accelyave*convert;

	FILE *writefile = fopen("/mnt/cf/cal.txt", "w");
	fprintf(writefile, "%4.6f\n", gyroave);			// Write to cal.txt
	fprintf(writefile, "%4.6f\n", accelxmax);
	fprintf(writefile, "%4.6f\n", accelxmin);
	fprintf(writefile, "%4.6f\n", accelxave);
	fprintf(writefile, "%4.6f\n", accelymax);
	fprintf(writefile, "%4.6f\n", accelymin);
	fprintf(writefile, "%4.6f\n", accelyave);


return 0;
}
```

Daq.cpp

```
#include <iostream>
#include <stdio.h>
#include <stdlib.h>
#include <fstream>
#include <cstdlib>
#include <sys/time.h>
#include <stddef.h>
	//stdio.h allows for use of System Function
	//fstream allows for I/O with Files
using namespace std;

int main()
{
	double initial, time, elapsed;
	struct timeval tp;
	int rtn;
	int x = 1;
	double average = 0;
	char d[10];
	FILE *file1;

	cout << "Data Acquisition Started" << endl;

	//User specified number of data points in datapoints.txt
	file1 = fopen("/mnt/cf/datapoints.txt", "r");
	fgets(d, 10, file1);
	int datasamples;
	datasamples = atoi(d);
	fclose(file1);
	cout << "Number of Data Points: " << datasamples << endl;



	//OPEN AND CLEAR FILES
	FILE *writefile1 = fopen("/mnt/cf/accelx.txt", "w");
	fclose(writefile1);
	FILE *writefile2 = fopen("/mnt/cf/accely.txt", "w");
	fclose(writefile2);
	FILE *writefile3 = fopen("/mnt/cf/gyro.txt", "w");
	fclose(writefile3);
	FILE *writefile7 = fopen("/mnt/cf/gyroave.txt", "w");
	fclose(writefile7);
	FILE *writefile8 = fopen("/mnt/cf/temp.txt", "w");
	fclose(writefile8);
	FILE *writefile = fopen("/mnt/cf/timing.txt", "w");

	//AVERAGING VARIABLES
```

```
	char c[10];  // declare a char array
	double data4 = 0;
	double avearray [15];
	int counter = 1;

	system("i2c-io 0x0b setdir g.3 out"); //Set Pin Direction
	system("i2c-io 0x0b set g.3 0"); //Blue LED on

	rtn = gettimeofday(&tp, NULL);
	initial = (double)tp.tv_sec+(1.e-6)*tp.tv_usec;

	//Get Data
	while(x <= datasamples)
		{

			rtn = gettimeofday(&tp, NULL);
			time = (double)tp.tv_sec+(1.e-6)*tp.tv_usec;
			elapsed = time - initial;
			//cout << elapsed << endl;
			fprintf(writefile, "%3.6f\n", elapsed);
			system("i2c-io 0x0b Get ADC.0 >> /mnt/cf/accelx.txt");
			system("i2c-io 0x0b Get ADC.1 >> /mnt/cf/accely.txt");
			system("i2c-io 0x0b Get ADC.2 >> /mnt/cf/gyro.txt");
			system("i2c-io 0x0b Get ADC.7 >> /mnt/cf/temp.txt");
			x++;
		}

	fclose(writefile);
	system("i2c-io 0x0b set g.3 1");                    //Blue LED off
	cout << "Completed" << endl;



/*


	//AVERAGING FOR ACCELX
	FILE *file;
	file = fopen("/mnt/cf/accelx.txt", "r");
	if(file==NULL)
	{
		printf("Error: can't open accelx file.\n");

		return 1;
	}
	else
	{
		printf("File accelx opened successfully.\n");
```

```
		double data4 = 0;
		double avearray [datasamples];
		int counter = 1;



		while(fgets(c, 10, file)!=NULL)
		{
			// keep looping until NULL pointer
			data4 = atof(c);
			avearray[counter] = data4;
			//cout << avearray[counter] << endl;
			//cout << data4 << endl;
			counter++;
		}
	}


	cout << "Averaging Accelx" << endl;
	FILE *writefile9 = fopen("/mnt/cf/accelxave.txt", "w");
	counter = 1;
	average = 0;
	while(counter <= datasamples)
	{
		if (counter == 1)
		{
			fprintf(writefile9, "%4.4f\n", avearray[1]);
		}
		if (counter == 2)
		{
			average = (avearray[counter]+avearray[counter-1])/2;
			fprintf(writefile9, "%4.4f\n", average);
		}
		if (counter == 3)
		{
			average = (avearray[counter]+avearray[counter-1]+avearray[counter-2])/3;
			fprintf(writefile9, "%4.4f\n", average);
		}
		if (counter >= 4)
		{
			average = (avearray[counter]+avearray[counter-1]+avearray[counter-2]+
avearray[counter-3])/4;
			fprintf(writefile9, "%4.4f\n", average);
		}
		counter++;
	}

	fclose(writefile9);
	fclose(file);
```

```
//AVERAGING FOR ACCELY
FILE *file2;
file2 = fopen("/mnt/cf/accely.txt", "r");
if(file2==NULL)
{
	 printf("Error: can't open accely file.\n");

	 return 1;
}
else
{
	printf("File accely opened successfully.\n");

	double data4 = 0;
	double avearray [datasamples];
	int counter = 1;

	while(fgets(c, 10, file2)!=NULL)
	{
		// keep looping until NULL pointer
		data4 = atof(c);
		avearray[counter] = data4;
		//cout << avearray[counter] << endl;
		//cout << data4 << endl;
		counter++;
	}
}

cout << "Averaging Accely" << endl;
FILE *writefile10 = fopen("/mnt/cf/accelyave.txt", "w");
counter = 1;
average = 0;
while(counter <= datasamples)
{
	if (counter == 1)
	{
		fprintf(writefile10, "%4.4f\n", avearray[1]);
	}
	if (counter == 2)
	{
		average = (avearray[counter]+avearray[counter-1])/2;
		fprintf(writefile10, "%4.4f\n", average);
	}
	if (counter == 3)
	{
		average = (avearray[counter]+avearray[counter-1]+avearray[counter-2])/3;
		fprintf(writefile10, "%4.4f\n", average);
	}
```

```
		if (counter >= 4)
		{
			average = (avearray[counter]+avearray[counter-1]+avearray[counter-2]+
avearray[counter-3])/4;
			fprintf(writefile10, "%4.4f\n", average);
		}
		counter++;
	}

	fclose(writefile10);
	fclose(file2);


//AVERAGING FOR GYRO
	FILE *file3;
	file3 = fopen("/mnt/cf/gyro.txt", "r");
	if(file3==NULL)
	{
		printf("Error: can't open gyro file.\n");

		return 1;
	 }
	else
	{
		 printf("File gyro opened successfully.\n");

		double data4 = 0;
		double avearray [datasamples];
		int counter = 1;



		while(fgets(c, 10, file3)!=NULL)
		{
			// keep looping until NULL pointer
			data4 = atof(c);
			avearray[counter] = data4;
			//cout << avearray[counter] << endl;
			//cout << data4 << endl;
			counter++;
		}
	}


	cout << "Averaging Gyro" << endl;
	FILE *writefile8 = fopen("/mnt/cf/gyroave.txt", "w");
	counter = 1;
	average = 0;
	while(counter <= datasamples)
```

```
	{
		if (counter == 1)
		{
			fprintf(writefile8, "%4.4f\n", avearray[1]);
		}
		if (counter == 2)
		{
			average = (avearray[counter]+avearray[counter-1])/2;
			fprintf(writefile8, "%4.4f\n", average);
		}
		if (counter == 3)
		{
			average = (avearray[counter]+avearray[counter-1]+avearray[counter-2])/3;
			fprintf(writefile8, "%4.4f\n", average);
		}
		if (counter >= 4)
		{
			average = (avearray[counter]+avearray[counter-1]+avearray[counter-2]+
avearray[counter-3])/4;
			fprintf(writefile8, "%4.4f\n", average);
		}
		counter++;
	}
	fclose(writefile8);
	fclose(file3);
*/

return 0;

}
```

Accelxave.cpp

```
#include <iostream>
#include <stdio.h>
#include <stdlib.h>
#include <fstream>
#include <cstdlib>
#include <sys/time.h>
```

```
#include <stddef.h>
#include <math.h>

using namespace std;

int main() {
  char c[10];  // declare a char array
  FILE *file;  // declare a FILE pointer
char d[10];
          FILE *file1;
          file1 = fopen("/mnt/cf/datapoints.txt", "r");
          fgets(d, 10, file1);
          int datasamples;
          datasamples = atoi(d);
          fclose(file1);
          //cout << "Number of Data Points: " << datasamples << endl;
          double data4 = 0;
          double avearray [datasamples];
          int counter = 1;
  file = fopen("/mnt/cf/accelxcal.txt", "r");


  if(file==NULL) {
    printf("Error: can't open accelxcal file.\n");

    return 1;
  }
  else {
    printf("File accelxcal opened successfully.\n");

          double data4 = 0;
          double avearray [datasamples];
          int counter = 1;



          while(fgets(c, 10, file)!=NULL) {
          // keep looping until NULL pointer
          //printf("String: %s", c);
          data4 = atof(c);
          avearray[counter] = data4;
          //cout << avearray[counter] << endl;
          //cout << data4 << endl;
          counter++;
          }

FILE *writefile1 = fopen("/mnt/cf/finaldata/accelxfinal.txt", "w");
counter = 1;
double average = 0;
```

```
while(counter <= datasamples){
        if (counter == 1){
        fprintf(writefile1, "%4.6f\n", avearray[1]);
        }
        if (counter == 2){
        average = (avearray[counter]+avearray[counter-1])/2;
        fprintf(writefile1, "%4.6f\n", average);
        }
        if (counter == 3){
        average = (avearray[counter]+avearray[counter-1]+avearray[counter-2])/3;
        fprintf(writefile1, "%4.6f\n", average);
        }
        if (counter >= 4){
        average = (avearray[counter]+avearray[counter-1]+avearray[counter-2]+ avearray[counter-
3])/4;
        fprintf(writefile1, "%4.6f\n", average);
        }
counter++;
}
}


//readFile1.close();
//fclose(readFile1);
//fclose(writefile1);

printf ("Completed\n");


return 0;
}
```

Accelyave.cpp

```
#include <iostream>
#include <stdio.h>
#include <stdlib.h>
#include <fstream>
#include <cstdlib>
#include <sys/time.h>
#include <stddef.h>
#include <math.h>

using namespace std;

int main() {
  char c[10];  // declare a char array
  FILE *file;  // declare a FILE pointer
char d[10];
```

```
        FILE *file1;
        file1 = fopen("/mnt/cf/datapoints.txt", "r");
        fgets(d, 10, file1);
        int datasamples;
        datasamples = atoi(d);
        fclose(file1);
        //cout << "Number of Data Points: " << datasamples << endl;
        double data4 = 0;
        double avearray [datasamples];
        int counter = 1;
 file = fopen("/mnt/cf/accelycal.txt", "r");


 if(file==NULL) {
  printf("Error: can't open accelycal file.\n");

  return 1;
 }
 else {
  printf("File accelycal opened successfully.\n");

        double data4 = 0;
        double avearray [datasamples];
        int counter = 1;



        while(fgets(c, 10, file)!=NULL) {
    // keep looping until NULL pointer
    //printf("String: %s", c);
        data4 = atof(c);
        avearray[counter] = data4;
        //cout << avearray[counter] << endl;
        //cout << data4 << endl;
        counter++;
        }

FILE *writefile1 = fopen("/mnt/cf/finaldata/accelyfinal.txt", "w");
counter = 1;
double average = 0;
while(counter <= datasamples){
        if (counter == 1){
        fprintf(writefile1, "%4.6f\n", avearray[1]);
        }
        if (counter == 2){
        average = (avearray[counter]+avearray[counter-1])/2;
        fprintf(writefile1, "%4.6f\n", average);
        }
        if (counter == 3){
```

```
        average = (avearray[counter]+avearray[counter-1]+avearray[counter-2])/3;
        fprintf(writefile1, "%4.6f\n", average);
        }
        if (counter >= 4){
        average = (avearray[counter]+avearray[counter-1]+avearray[counter-2]+ avearray[counter-
3])/4;
        fprintf(writefile1, "%4.6f\n", average);
        }
counter++;
}
}


//readFile1.close();
//fclose(readFile1);
//fclose(writefile1);

printf ("Completed\n");


return 0;
}
```

Gyroave.cpp

```
#include <iostream>
#include <stdio.h>
#include <stdlib.h>
#include <fstream>
#include <cstdlib>
#include <sys/time.h>
#include <stddef.h>
#include <math.h>

using namespace std;

int main() {
  char c[10];  // declare a char array
  FILE *file;  // declare a FILE pointer
        char d[10];
        FILE *file1;
        file1 = fopen("/mnt/cf/datapoints.txt", "r");
        fgets(d, 10, file1);
        int datasamples;
        datasamples = atoi(d);
        fclose(file1);
        //cout << "Number of Data Points: " << datasamples << endl;
        double data4 = 0;
```

```
        double avearray [datasamples];
        int counter = 1;

  file = fopen("/mnt/cf/gyrocal.txt", "r");


  if(file==NULL) {
    printf("Error: can't open gyrocal file.\n");

    return 1;
  }
  else {
    printf("File gyrocal opened successfully.\n");
        counter = 1;

        while(fgets(c, 11, file)!=NULL) {
      // keep looping until NULL pointer
      //printf("String: %s", c);
        data4 = atof(c);
        avearray[counter] = data4;
        //cout << avearray[counter] << endl;
        //cout << data4 << endl;
        //cout << counter << endl;
        counter++;
        }
        cout << "Done" << endl;
FILE *writefile1 = fopen("/mnt/cf/finaldata/gyrofinal.txt", "w");
counter = 1;
double average = 0;
while(counter <= datasamples){
        if (counter == 1){
        fprintf(writefile1, "%4.6f\n", avearray[1]);
        }
        if (counter == 2){
        average = (avearray[counter]+avearray[counter-1])/2;
        fprintf(writefile1, "%4.6f\n", average);
        }
        if (counter == 3){
        average = (avearray[counter]+avearray[counter-1]+avearray[counter-2])/3;
        fprintf(writefile1, "%4.6f\n", average);
        }
        if (counter >= 4){
        average = (avearray[counter]+avearray[counter-1]+avearray[counter-2]+ avearray[counter-
3])/4;
        fprintf(writefile1, "%4.6f\n", average);
        }
counter++;
}
}
```

```
printf ("Completed\n");


return 0;
}
```

Finalcalcx.cpp

```
#include <iostream>
#include <stdio.h>
#include <stdlib.h>
#include <fstream>
#include <cstdlib>
#include <sys/time.h>
#include <stddef.h>
#include <math.h>

using namespace std;

int main()
{
	char c[10];  // declare a char array
	FILE *file;  // Pointer for accelx file
	FILE *file2;  // Pointer for cal file
	char d[10];
	FILE *file1;
	file1 = fopen("/mnt/cf/datapoints.txt", "r");
	fgets(d, 10, file1);
	int datasamples;
	datasamples = atoi(d);
	fclose(file1);
	double data = 0;
	double accelxfinal [datasamples];
	double caldata [7];
	int counter = 1;
	double convert = .004887585533;

	//Accelerometer X Ave Data
	 file = fopen("/mnt/cf/accelx.txt", "r");


	if(file==NULL)
	{
		printf("Error: can't open accelx file.\n");
		return 1;
	}

	else
```

```
{
        printf("File accelx opened successfully.\n");
        double data = 0;
        double accelxfinal [datasamples];
        int counter = 1;

        while(fgets(c, 10, file)!=NULL)
        {
                // keep looping until NULL pointer
                //printf("String: %s", c);
                data = atof(c);
                accelxfinal[counter] = data;
                //cout << accelxfinal[counter] << endl;
                //cout << data << endl;
                counter++;
        }


//Getting Calibration Data
        file2 = fopen("/mnt/cf/cal.txt", "r");


        if(file2==NULL)
        {
                printf("Error: can't open cal file.\n");

                return 1;
        }

        else
        {
                printf("File cal opened successfully.\n");

                double data = 0;
                double caldata [datasamples];
                int counter = 1;

                while(fgets(c, 11, file2)!=NULL)
                {
                        data = atof(c);
                        caldata[counter] = data;
                        //cout << caldata[counter] << endl;
                        counter++;
                }

                //Writing Accelerometer X Final Data
                FILE *writefile = fopen("/mnt/cf/accelxcal.txt", "w");
                counter = 1;
```

```
                        while(counter <= datasamples)
                        {
                                accelxfinal[counter] = ((accelxfinal[counter] * convert) - caldata[4]);
                                fprintf(writefile, "%4.6f\n", accelxfinal[counter]);
                                counter++;
                        }
                        fclose(writefile);


                }
                printf ("Completed\n");
        }

return 0;
}
```

Finalcaly.cpp

```
#include <iostream>
#include <stdio.h>
#include <stdlib.h>
#include <fstream>
#include <cstdlib>
#include <sys/time.h>
#include <stddef.h>
#include <math.h>

using namespace std;

int main()
{
        char c[10];  // declare a char array
        FILE *file;  // Pointer for accely file
        FILE *file2;  // Pointer for cal file
        char d[10];
        FILE *file1;
        file1 = fopen("/mnt/cf/datapoints.txt", "r");
        fgets(d, 10, file1);
        int datasamples;
        datasamples = atoi(d);
        fclose(file1);
        double data = 0;
        double accelyfinal [datasamples];
        double caldata [7];
        int counter = 1;
        double convert = .004887585533;

        //Accelerometer X Ave Data
         file = fopen("/mnt/cf/accely.txt", "r");
```

```
if(file==NULL)
{
	printf("Error: can't open accely file.\n");
	return 1;
}

else
{
	printf("File accely opened successfully.\n");
	double data = 0;
	double accelyfinal [datasamples];
	int counter = 1;

	while(fgets(c, 10, file)!=NULL)
	{
		// keep looping until NULL pointer
		//printf("String: %s", c);
		data = atof(c);
		accelyfinal[counter] = data;
		//cout << accelyfinal[counter] << endl;
		//cout << data << endl;
		counter++;
	}


//Getting Calibration Data
	file2 = fopen("/mnt/cf/cal.txt", "r");


	if(file2==NULL)
	{
		printf("Error: can't open cal file.\n");

		return 1;
	}

	else
	{
		 printf("File cal opened successfully.\n");

		double data = 0;
		double caldata [datasamples];
		int counter = 1;

		while(fgets(c, 11, file2)!=NULL)
		{
			data = atof(c);
```

```
                caldata[counter] = data;
                //cout << caldata[counter] << endl;
                counter++;
        }

        //Writing Accelerometer Y Final Data
        FILE *writefile = fopen("/mnt/cf/accelycal.txt", "w");
        counter = 1;

        while(counter <= datasamples)
        {
                accelyfinal[counter] = ((accelyfinal[counter] * convert) - caldata[7]);
                fprintf(writefile, "%4.6f\n", accelyfinal[counter]);
                counter++;
        }
        fclose(writefile);


    }
    printf ("Completed\n");
  }

return 0;
}
```

Finalcalcg.cpp

```
#include <iostream>
#include <stdio.h>
#include <stdlib.h>
#include <fstream>
#include <cstdlib>
#include <sys/time.h>
#include <stddef.h>
#include <math.h>

using namespace std;
int main()
{
        char c[10];  // declare a char array
        FILE *file;  // Pointer for gyroave file
        FILE *file2;  // Pointer for cal file
        char d[10];
        int datasamples;
        FILE *file1;
        file1 = fopen("/mnt/cf/datapoints.txt", "r");
        fgets(d, 10, file1);
        datasamples = atoi(d);
        fclose(file1);
```

```
double data = 0;
double gyrofinal1 [datasamples];
double caldata [7];
int counter = 1;
double convert = .004887585533;

//Open Gyro Data
 file = fopen("/mnt/cf/gyro.txt", "r");

if(file==NULL)
{
	printf("Error: can't open gyro file.\n");
	return 1;
}

else
{
	printf("File gyro opened successfully.\n");
	//double data = 0;
	//double gyrofinal [datasamples];
	//int counter = 1;

	while(fgets(c, 10, file)!=NULL)
	{
		// keep looping until NULL pointer
		//printf("String: %s", c);
		data = atof(c);
		cout << data << endl;
		gyrofinal1[counter] = data;
		cout << gyrofinal1[counter] << endl;
		counter++;
	}


//Getting Calibration Data
	file2 = fopen("/mnt/cf/cal.txt", "r");


	if(file2==NULL)
	{
		printf("Error: can't open cal file.\n");

		return 1;
	}

	else
	{
		 printf("File cal opened successfully.\n");
```

```
                        double data = 0;
                        double caldata [datasamples];
                        int counter = 1;

                        while(fgets(c, 10, file2)!=NULL)
                        {
                                data = atof(c);
                                caldata[counter] = data;
                                //cout << caldata[counter] << endl;
                                counter++;
                        cout << gyrofinal1[counter-1] << endl;
                        /*cout << gyrofinal1[2] << endl;
                        cout << gyrofinal1[3] << endl;
                        cout << gyrofinal1[4] << endl;*/
                        }
                        cout << "hi" << endl;

                        //Writing Gyroscope Final Data
                        FILE *writefile = fopen("/mnt/cf/gyrocal.txt", "w");
                        counter = 1;



                        while(counter <= datasamples)
                        {
                                cout << "Data point #:" << counter << endl;
                                cout << gyrofinal1[counter] << endl;

                                gyrofinal1[counter] = ((gyrofinal1[counter] * convert) - caldata[1]);
                                cout << gyrofinal1[counter] << endl;
                                fprintf(writefile, "%4.6f\n", gyrofinal1[counter]);
                                counter++;
                        }
                        //fclose(writefile);


                }
                printf ("Completed\n");
        }

return 0;
}
```

Positionx.cpp

```
#include <iostream>
#include <stdio.h>
#include <stdlib.h>
```

```
#include <fstream>
#include <cstdlib>
#include <sys/time.h>
#include <stddef.h>
#include <math.h>

using namespace std;

int main()
{
	char c[10];  // declare a char array
	FILE *file;  // Pointer for accelxcal file
	FILE *file2;  // Pointer for timing file
	char d[10];
	FILE *file1;
	file1 = fopen("/mnt/cf/datapoints.txt", "r");
	fgets(d, 10, file1);
	int datasamples;
	datasamples = atoi(d);
	fclose(file1);
	double data = 0;
	double position [datasamples];
	int counter = 1;

	//Getting Accelerometer Data

	//file = fopen("/mnt/cf/accelxcal.txt", "r");
	file = fopen("/mnt/cf/accelxcal.txt", "r");

	if(file==NULL)
	{
		printf("Error: can't open accelxfinal file.\n");
		return 1;
	}

	else
	{
		printf("File accelxfinal opened successfully.\n");
		double data = 0;
		double position [datasamples];
		int counter = 1;

		while(fgets(c, 11, file)!=NULL)
		{
			// keep looping until NULL pointer
			//printf("String: %s", c);
			data = atof(c);
			position[counter] = data;
			//cout << position[counter] << endl;
```

```
		//cout << data << endl;
		counter++;
	}


//Getting Timing Data
	file2 = fopen("/mnt/cf/timing.txt", "r");


	if(file2==NULL)
	{
		printf("Error: can't open timing file.\n");

		return 1;
	}

	else
	{
		 printf("File timing opened successfully.\n");

		double data = 0;
		double timing [datasamples];
		int counter = 1;

		while(fgets(c, 11, file2)!=NULL)
		{
			data = atof(c);
			timing[counter] = data;
			//cout << timing[counter] << endl;
			counter++;
		}
	fclose(file2);
		//Writing Accelx Positioning Data
		FILE *writefile = fopen("/mnt/cf/positionx.txt", "w");
		counter = 1;

		//double acceleration = 0; //keeps track of current acceleration
		double velocity = 0; //keep track of current velocity

		while(counter <= datasamples)
		{

			if (counter < datasamples)
			{
			//Acceleration Calculation

				if (position[counter] <= 0 && position[counter] >= 0)
					{
						position[counter] = 0;
```

```
                                        }
                                else
                                        {
                                                //Do not change
                                        }


                                position[counter] = position[counter]*9.8;
                                //acceleration = acceleration + position[counter];
                                //position[counter] = acceleration;
                                //Position Calculation
                                //First Integration (Velocity)
                                position[counter] = ((position[counter]*timing[counter+1])-
(position[counter]*timing[counter]));

                                velocity = velocity + position[counter];
                                        position[counter] = velocity;

                                //Second Integration (Distance)

                                position[counter] = ((position[counter]*timing[counter+1])-
(position[counter]*timing[counter]));
                                fprintf(writefile, "%4.4f\n", position[counter]);


                                }
                                else
                                {
                                        position[counter] = 0;
                                        fprintf(writefile, "%4.4f\n", position[counter]);
                                }

                        counter++;
                        }


                }
                printf ("Completed\n");
        }

return 0;
}
```

Positiony.cpp

```
#include <iostream>
#include <stdio.h>
#include <stdlib.h>
#include <fstream>
```

```
#include <cstdlib>
#include <sys/time.h>
#include <stddef.h>
#include <math.h>

using namespace std;

int main()
{
	char c[10];  // declare a char array
	FILE *file;  // Pointer for accelycal file
	FILE *file2;  // Pointer for timing file
	char d[10];
	FILE *file1;
	file1 = fopen("/mnt/cf/datapoints.txt", "r");
	fgets(d, 10, file1);
	int datasamples;
	datasamples = atoi(d);
	fclose(file1);
	double data = 0;
	double position [datasamples];
	int counter = 1;

	//Getting Accelerometer Data
	 //file = fopen("/mnt/cf/finaldata/accelyfinal.txt", "r");
	file = fopen("/mnt/cf/accelycal.txt", "r");


	if(file==NULL)
	{
		printf("Error: can't open accelyfinal file.\n");
		return 1;
	}

	else
	{
		printf("File accelyfinal opened successfully.\n");
		double data = 0;
		double position [datasamples];
		int counter = 1;

		while(fgets(c, 11, file)!=NULL)
		{
			// keep looping until NULL pointer
			//printf("String: %s", c);
			data = atof(c);
			position[counter] = data;
			//cout << position[counter] << endl;
			//cout << data << endl;
```

```
		counter++;
	}


//Getting Timing Data
	file2 = fopen("/mnt/cf/timing.txt", "r");


	if(file2==NULL)
	{
		printf("Error: can't open timing file.\n");

		return 1;
	}

	else
	{
		 printf("File timing opened successfully.\n");

		double data = 0;
		double timing [datasamples];
		int counter = 1;

		while(fgets(c, 11, file2)!=NULL)
		{
			data = atof(c);
			timing[counter] = data;
			//cout << timing[counter] << endl;
			counter++;
		}
	fclose(file2);
		//Writing Accely Positioning Data
		FILE *writefile = fopen("/mnt/cf/positiony.txt", "w");
		counter = 1;

		//double acceleration = 0; //keeps track of current acceleration
		double velocity = 0; //keep track of current velocity

		while(counter <= datasamples)
		{

			if (counter < datasamples)
			{

				if (position[counter] <= 0 && position[counter] >= 0)
					{
						position[counter] = 0;
					}
				else
```

```
                                        {
                                        //Do Not Change
                                        }
                                //Acceleration Calcuate
                                position[counter] = position[counter]*9.8;
                                //acceleration = acceleration + position[counter];
                                //position[counter] = acceleration;

                                //Position Calculation
                                //First derivative (Velocity)
                                position[counter] = ((position[counter]*timing[counter+1])-
(position[counter]*timing[counter]));

                                velocity = velocity + position[counter];
                                position[counter] = velocity;

                                //Second Derivative (Distance)
                                position[counter] = ((position[counter]*timing[counter+1])-
(position[counter]*timing[counter]));
                                fprintf(writefile, "%4.4f\n", position[counter]);

                        }
                        else
                        {
                        position[counter] = 0;
                        fprintf(writefile, "%4.4f\n", position[counter]);
                        }
                        counter++;
                }


        }
        printf ("Completed\n");
    }

return 0;
}
```

Positiong.cpp

```
#include <iostream>
#include <stdio.h>
#include <stdlib.h>
#include <fstream>
#include <cstdlib>
#include <sys/time.h>
#include <stddef.h>
#include <math.h>
```

```
using namespace std;

int main()
{
	char c[10];  // declare a char array
	FILE *file;  // Pointer for gyrocalcal file
	FILE *file2;  // Pointer for timing file
	char d[10];
	FILE *file1;
	file1 = fopen("/mnt/cf/datapoints.txt", "r");
	fgets(d, 10, file1);
	int datasamples;
	datasamples = atoi(d);
	fclose(file1);
	double data = 0;
	double position [datasamples];
	int counter = 1;

	//Getting Gyro Data
	 //file = fopen("/mnt/cf/finaldata/gyrofinal.txt", "r");
	file = fopen("/mnt/cf/gyrocal.txt", "r");


	if(file==NULL)
	{
		printf("Error: can't open gyrofinal file.\n");
		return 1;
	}

	else
	{
		printf("File gyrofinal opened successfully.\n");
		double data = 0;
		double position [datasamples];
		int counter = 1;

		while(fgets(c, 11, file)!=NULL)
		{
			// keep looping until NULL pointer
			//printf("String: %s", c);
			data = atof(c);
			position[counter] = data;
			//cout << position[counter] << endl;
			//cout << data << endl;
			counter++;
		}


	//Getting Timing Data
```

```
file2 = fopen("/mnt/cf/timing.txt", "r");


if(file2==NULL)
{
	printf("Error: can't open timing file.\n");

	return 1;
}

else
{
	 printf("File timing opened successfully.\n");

	double data = 0;
	double timing [datasamples];
	int counter = 1;

	while(fgets(c, 11, file2)!=NULL)
	{
		data = atof(c);
		timing[counter] = data;
		//cout << timing[counter] << endl;
		counter++;
	}
fclose(file2);
	//Writing Gyro Degree Data
	FILE *writefile = fopen("/mnt/cf/positiong.txt", "w");
	counter = 1;

	double degrees = 0; //keeps track of current degree status
	while(counter <= datasamples)
	{

		if (counter < datasamples)
		{
		position[counter] = ((position[counter]/.0125)*(timing[counter+1]-
timing[counter]));
		degrees = degrees + position[counter];
		position[counter] = degrees;
		fprintf(writefile, "%4.6f\n", position[counter]);
		}
		else
		{
		position[counter] = degrees;
		fprintf(writefile, "%4.6f\n", position[counter]);
		}
		counter++;
	}
```

```
                }
                printf ("Completed\n");
        }

return 0;
}
```

Finalpos.cpp

```
#include <iostream>
#include <stdio.h>
#include <stdlib.h>
#include <fstream>
#include <cstdlib>
#include <sys/time.h>
#include <stddef.h>
#include <math.h>
#define Pi 3.141592654

using namespace std;

int main()
{
        char c[10];  // declare a char array
        FILE *file;  // Pointer for positionx file
        FILE *file2;  // Pointer for positiony file
        FILE *file3; // Pointer for positiong file
        char d[10];
        FILE *file1;
        file1 = fopen("/mnt/cf/datapoints.txt", "r");
        fgets(d, 10, file1);
        int datasamples;
        datasamples = atoi(d);
        fclose(file1);
        double positionx [datasamples];
        double positiony [datasamples];
        double positiong [datasamples];
        int counter = 1;
        double rtod = .01745329252;

//Getting X-axis Data
        file = fopen("/mnt/cf/positionx.txt", "r");
        if(file==NULL)
        {
                printf("Error: can't open positionx file.\n");
                return 1;
        }
```

```
	else
	{
		printf("File positionx opened successfully.\n");
		double data = 0;
		int counter = 1;
		while(fgets(c, 12, file)!=NULL)
		{
			// keep looping until NULL pointer
			//printf("String: %s", c);
			data = atof(c);
			positionx[counter] = data;
			//cout << position[counter] << endl;
			//cout << data << endl;
			counter++;
		}
	}

//Getting Y-axis Data
	file = fopen("/mnt/cf/positiony.txt", "r");
	if(file==NULL)
	{
		printf("Error: can't open positiony file.\n");
		return 1;
	}
	else
	{
		printf("File positiony opened successfully.\n");
		double data = 0;
		int counter = 1;
		while(fgets(c, 12, file)!=NULL)
		{
			// keep looping until NULL pointer
			//printf("String: %s", c);
			data = atof(c);
			positiony[counter] = data;
			//cout << position[counter] << endl;
			//cout << data << endl;
			counter++;
		}
	}

//Getting Gyro Data
	file = fopen("/mnt/cf/positiong.txt", "r");
	if(file==NULL)
	{
		printf("Error: can't open positiong file.\n");
		return 1;
	}
	else
```

```
	{
		printf("File positiong opened successfully.\n");
		double data = 0;
		int counter = 1;
		while(fgets(c, 12, file)!=NULL)
		{
			// keep looping until NULL pointer
			//printf("String: %s", c);
			data = atof(c);
			positiong[counter] = data;
			fprintf(writefile5, "%4.6f\n", positiong[counter]);
			//cout << position[counter] << endl;
			//cout << data << endl;
			counter++;
		}
	}

//Positioning Calculations

	counter = 1;
	double theta = 0;
	double phi = 0;
	double xpos = 0;
	double ypos = 0;
	double pvectorx = 0;
	double pvectory = 0;
	double xfinalpos = 0;
	double yfinalpos = 0;

	//Open files for writing position
	FILE *writefile = fopen("/mnt/cf/xposition.txt", "w");
	FILE *writefile1 = fopen("/mnt/cf/yposition.txt", "w");
	FILE *writefile2 = fopen("/mnt/cf/phi.txt", "w");
	FILE *writefile3 = fopen("/mnt/cf/xvector.txt", "w");
	FILE *writefile4 = fopen("/mnt/cf/yvector.txt", "w");
	FILE *writefile6 = fopen("/mnt/cf/xpos.txt", "w");
	FILE *writefile7 = fopen("/mnt/cf/ypos.txt", "w");
	while(counter <= datasamples)
		{
			//how to fix positionx[counter] = 0???????
			if (positionx[counter] == 0)
			{
				if (positiong[counter] == 0 && positiony[counter] < 0)
				{
					theta = (-Pi/2);
					//cout << "Option 1" << endl;
				}
				else if (positiong[counter] == 0 && positiony[counter] > 0)
				{
```

```
                        theta = (Pi/2);
                        //cout << "Option 2" << endl;
                }
                else if (positiong[counter] < 0 && positiony[counter] < 0)
                {
                        theta = (-Pi/2);
                        //cout << "Option 3" << endl;
                }
                else if (positiong[counter] < 0 && positiony[counter] > 0)
                {
                        theta = (Pi/2);
                        //cout << "Option 4" << endl;
                }
                else if (positiong[counter] > 0 && positiony[counter] < 0)
                {
                        theta = (-Pi/2);
                        //cout << "Option 5" << endl;
                }
                else if (positiong[counter] > 0 && positiony[counter] > 0)
                {
                        theta = (Pi/2);
                        //cout << "Option 6" << endl;
                }
                else
                {
                        theta = 0;
                        //cout << "Option 7" << endl;
                }
        }

        else
        {
                theta = atan(positiony[counter]/positionx[counter]);
                //cout << "Option 8" << endl;
        }

        phi = theta + (positiong[counter]*rtod);
        fprintf(writefile2, "%4.6f\n", phi);
        //cout << phi << endl;
        pvectorx =
sqrt(positionx[counter]*positionx[counter]+positiony[counter]*positiony[counter]);
        pvectory = pvectorx;
        if (positionx[counter] < 0)
        {
                pvectorx = -1*pvectorx;
                pvectory = -1*pvectory;
                //cout << "Option 9" << endl;
        }
```

```
                else
                {
                        //pvectorx = pvectorx;
                        //cout << "Option 10" << endl;
                }

                xpos = pvectorx*cos(phi);
                ypos = pvectory*sin(phi);

                fprintf(writefile3, "%4.6f\n", pvectorx);
                fprintf(writefile4, "%4.6f\n", pvectory);
                fprintf(writefile6, "%4.6f\n", xpos);
                fprintf(writefile7, "%4.6f\n", ypos);

                xfinalpos = xfinalpos + xpos;
                yfinalpos = yfinalpos + ypos;

                //Writing X Position Data
                fprintf(writefile, "%4.6f\n", xfinalpos);
                //Writing Y Position Data
                fprintf(writefile1, "%4.6f\n", yfinalpos);

                counter++;


        }

        cout << "-------------------------------------" << endl;
        cout << "Final Coordinates:" << endl;
        cout << "X-axis: " << xfinalpos << " meters" << endl;
        cout << "Y-axis: " << yfinalpos << " meters" << endl;
        cout << "-------------------------------------" << endl;

return 0;
}
```